UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/826,085	04/04/2001	Matthew Bunkley Trevathan	RSW920010072US1	3871

7590 02/24/2004

SCHMEISER OLSEN & WATTS
3 LEAR JET LANE SUITE 201
LATHAM, NY 12110

EXAMINER	
CHOI, WOO H	

ART UNIT	PAPER NUMBER
2186	13

DATE MAILED: 02/24/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
P.O. BOX 1450
ALEXANDRIA, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Paper No. 13

Application Number: 09/826,085
Filing Date: April 04, 2001
Appellant(s): TREVATHAN, MATTHEW BUNKLEY

Jack P. Friedman
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed November 17.

(1) Real Party in Interest

A statement identifying the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

A statement identifying the related appeals and interferences which will directly affect or be directly affected by or have a bearing on the decision in the pending appeal is contained in the brief.

(3) Status of Claims

The statement of the status of the claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Invention

The summary of invention contained in the brief is correct.

(6) Issues

The appellant's statement of the issues in the brief is correct.

(7) Grouping of Claims

Appellant's brief includes a statement that claims 1, 3 – 9 do not stand or fall together and provides reasons as set forth in 37 CFR 1.192(c)(7) and (c)(8).

The copy of the appealed claims contained in the Appendix to the brief is correct.

(9) Prior Art of Record

6,272,598	Arlitt et al.	08-2001
6,223,256	Gaither	04-2001

(10) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

1. Claims 1, 3 – 7, and 9 are rejected under 35 U.S.C. 102(e) as being anticipated by Gaither (US Patent No. 6,223,256).

2. With respect to claim 1, Gaither discloses a method for managing a cache (abstract), comprising the acts of:

analyzing information stored in a caching profile (col. 8, lines 41 – 47, program run-time behavior information is gathered and analyzed to assign cache class attributes. Information gathered, i.e. 'cache profile' since the information pertains to the run-time caching behavior of a program, must be retained or stored for analysis and assignment of cache class attributes.); and

responsive to the act of analyzing, selecting a preferred caching algorithm from a plurality of caching algorithms (abstract, the analyzed information, or the cache class attribute is used to select a replacement algorithm);

wherein the act of analyzing is performed by a predictive modeling engine (col. 8, lines 45 – 47, collection and analysis of information is done by the operating system or other run-tim software. Col. 8, lines 41 – 42 cited above specifically states "dynamic analysis of program run-time behavior is used to assign cache class attributes." Abstract states "Class attributes may indicate a relative likelihood of future use." Since the collection of information computed and gathered, i.e. frequency or infrequency of entries, entry residency time, and replacement history, is a model of the run-time behavior of a program and since this model, or run-time behavior information, is used to assign class attributes that may indicate relative likelihood of future use, i.e. prediction of future use, the 'predictive modeling engine' is disclosed).

3. With respect to claims 3 and 4, Gaither discloses a method for managing a cache, comprising the acts of:

updating a caching profile in response to arrival of a file (col. 8 lines 41 – 53, act of updating a caching profile in response to arrival of a file is disclosed, since the collection and analysis of information is done *dynamically* at run-time. A program that is compiled is stored in one or more files. Run-time caching behavior information of a program cannot be collected until the program, i.e. file, arrives at a computer memory and more specifically at a cache. The collection of information gathered constitutes a 'caching profile' since it contains caching

behavior information. Gathering of information requires constant updating of this 'caching profile' as new information is generated and gathered while the program runs);

responsive to the acts of updating, analyzing information stored in the caching profile (as noted above, information collection and analyses are done dynamically. Analyzing information is necessarily responsive to the act of updating, i.e. information gathering, since the analysis cannot be done without the information to be analyzed. Dynamic analysis implies that the analysis is done 'on the fly' as the information is updated); and

responsive to the act of analyzing, selecting a preferred caching algorithm from a plurality of caching algorithms (abstract, the analyzed information, or the cache class attribute is used to select a replacement algorithm);

wherein the act of analyzing is performed by a predictive modeling engine (see the discussion of this limitation in claim 1 above).

4. With respect to claim 5, Gaither discloses a method for managing a cache, comprising the acts of:

responsive to arrival of a file at a cache (see the discussion of arrival of a compiled program above), analyzing information stored in a caching profile (i.e. dynamic analysis of gathered information) by computing a plurality of metrics (col. 5 lines 7 – 8, classes are ranked in a hierarchy, this requires some basis for comparison, or metrics. See also col. 8, lines 2 – 9, measure of likelihood of future use is also taught. This measure of likelihood comprises the metrics that are computed and gathered, i.e. frequency and infrequency of entries, residency

time, and cache history, see col. 8, lines 41 – 53. Determination of relative frequencies and residency time requires computing); and

responsive to a comparison of the metrics one with another, selecting preferred caching algorithm from a plurality of caching algorithms (col. 5, lines 14 – 15, also see above, see also title).

5. With respect to claims 6 and 7, the plurality of metrics includes clustering metrics and scattering metrics, respectively (col. 13, line 4, non-uniform distribution implies clustering. Col. 13, line 3, in a uniform distribution, things are scattered evenly. Metrics, i.e. frequent and infrequent entries, residency time, and replacement history, are analyzed to determine class attributes that are used in selecting the caching algorithms. Other than their association with certain algorithms, Applicant does not define what clustering and scattering metrics are. Therefore, a set of metrics that results in the selection of a caching algorithm that exhibit non-uniform distribution or clustering behavior can properly be called 'clustering metrics'. Likewise, a set of metrics that results in the selection of a caching algorithm that exhibit uniform distribution or scattering behavior can properly be called 'scattering metrics'. In addition, these metrics indeed are clustering and scattering metrics when interpreted in light of the Applicant's specification. The specification discloses that metrics for most used and least used caching algorithms are examples of measures of clustering and scattering, respectively, see page 8, lines 4 – 5 and page 10, lines 8 – 11. These are measures of relative frequencies. Metrics that are gathered and used in Gaither include **frequent** entries, i.e. most used or clustering metrics, and **infrequent** entries, i.e. least used or scattering metrics).

6. With respect to claim 9, the act of analyzing is performed by a predictive modeling engine (See discussion of claim 1 above).

Claim Rejections - 35 USC § 103

7. Claim 8 is rejected under 35 U.S.C. 103(a) as being unpatentable over Arlitt *et al.* (US Patent No. 6,272,598, hereinafter "Arlitt") in view of the Applicant's admitted prior art.

Arlitt discloses all of the limitations of the parent claim, claim 5, as follows:

Arlitt discloses a method for managing a cache, comprising the acts of:

responsive to arrival of a file at a cache, analyzing information stored in a caching profile by computing a plurality of metrics (col. 5, lines 48 – 50, **arrival** of a file is inherent since the subject matter of Arlitt's invention is a Web cache which caches Web page documents, see col. 1, lines 15 – 29. Hit rate and byte hit rate computations are necessarily responsive to arrival of a file. A hit rate computation can only begin in response to arrival of a file, since the reference to the file prior to its arrival is classified a 'miss', see col. 1, lines 53 – 54. Determination of hit rate and byte hit rate, i.e. 'cache profile', requires computations and storage.); and

responsive to a comparison of the metrics (i.e. hit rate and byte hit rate) one with another, selecting preferred caching algorithm from a plurality of caching algorithms (col. 5, lines 35 – 47, Arlitt discloses that "Each replacement policy is optimized for one performance metric." Arlitt makes it clear that the use of a particular replacement policy, i.e. caching algorithm, is

based on the performance metric. Algorithm selections based on the performance metrics require comparison of metrics).

Arlitt also discloses the following limitation:

wherein the plurality of caching algorithms includes a least-used caching algorithm (col. 6, lines 4 – 5) and a least-recently-used caching algorithm (col. 5, lines 62 – 63).

However, Arlitt does not specifically disclose the use of a most-recently-used caching algorithm and a most-used caching algorithm. On the other hand, Applicant admitted that these two algorithms were known on pages 2 and 3 of the specification.

It would have been obvious to one of ordinary skill in the art, having the teachings of Arlitt and the admitted prior art before him at the time the invention was made, to use the additional replacement algorithm teachings of the admitted prior art in the dynamic cache management system of Arlitt, in order to increase the versatility and performance of a web caching system since the additional algorithms allow the cache to adapt to caching patterns that not have been handled in an optimal way because of the limited number of caching options available prior to the combination. Also note that Arlitt specifically discloses that other known cache replacement algorithms may also be used (col. 7, lines 28 – 29).

(11) Response to Argument

Issue 1 – Claims 1, 3 – 7, and 9 are anticipated by Gaither (US Patent No. 6,223,256).

Claim 1

Summary of Appellant's Argument

> Gaither does not teach "wherein the act of **analyzing** is performed by a predictive modeling engine".
>
> Appellant contends that the Examiner's argument that the operating system or software performs the act of **analyzing** the information is not persuasive because all that is disclosed in col.8, lines 45 – 47 of Gaither is that the operating system or software gathers information and most certainly does not teach that the operating system or software analyzes information, as required by claim 1. (Appeal Brief, page 5, lines 16 – 20)

Examiner's Response

The examiner cited column 8, lines 41 – 47, on the first appearance of the limitation "**analyzing**" in the claim, where Gaither discloses the following:

> In a third option, ***dynamic analysis*** of program run-time behavior is used to assign cache class attributes. For example, the cache may be periodically observed to detect entries that are frequent or infrequent, entry residency time, and replacement history information. The information may be gathered by the operating system or by other run time software. (emphasis added)

The first sentence of the cited passage specifically states that "***dynamic analysis*** of program run-time behavior is used to assign cache class attributes." Dynamic analysis of program run-time behavior, which is modeled with metrics or measures that are gathered (i.e. frequency or infrequency of entries, entry residency time, and replacement history) implies that

the analysis is performed while the program is running, a point which Appellant later concedes (Brief, page 7, lines 8 – 13). Gaither specifically discloses, in col. 8, lines 51 – 53, that "the ***run-time software*** may ***dynamically*** change the cache class of pages (page-table-entries) based on run-time behavior." How is the run-time software to know how to change and what to change to ***dynamically*** if it merely gathers information? On col. 4, lines 60 – 62, Gaither further states that "the cache attribute may be determined at run-time based on historical data." This sentence further indicates analysis by the run-time software or the operating system.

The abstract states "Class attribute may indicate a relative likelihood of future use." The run-time behavior information (i.e. entry frequency or infrequency, entry residency time, cache replacement history) or model is used to gauge or predict future behavior, i.e. relative likelihood of future use. Therefore, this teaching, in conjunction with the modeling teaching of the program run-time behavior discussed above, reads on the limitation 'predictive modeling engine' as claimed and disclosed by Appellant.

Claim 3

Appellant presents three separate arguments, each of which are summarized and responded to individually.

Summary of Appellant's Argument 1

Gaither does not teach "updating a **caching profile**".

The only updating that is disclosed in col. 8, lines 41 – 53 of Gaither is that "the run-time software may *dynamically* **change** the cache class of pages (page-table-entries) based on run-time behavior" (emphasis added) (Appeal Brief, page 6, lines 15 – 18).

The Examiner has not even identified the caching profile that is updated. (Appeal Brief, page 6, line 21)

The Examiner notes that Appellant apparently agrees with the Examiner's position presented above with respect to claim 1, since dynamic change of cache class based on run-time behavior, by the run-time software, requires analysis or examination of the run-time behavior.

Examiner's Response to Argument 1

Gaither discloses gathering of run-time behavioral information related to caching, such as frequency, residency time, and replacement history (col. 8, lines 42 – 45). Based on the collected information, cache attributes are assigned (col. 8, lines 41 – 42) or changed (col. 8, lines 51 – 53). While Gaither does not call this collection of information a 'caching profile', the collection contains run-time caching behavioral information (i.e. 'caching profile'). This collection of information (i.e. caching profile) is updated as new information is collected and added to the set of previously collected information while the program is running. Further, "Dynamic analysis" implies updating of this caching profile.

Summary of Appellant's Argument 2

Gaither does not teach 'updating a caching profile **in response to arrival of a file**".

> Gaither does not teach that data generated and processed (from execution of program code) is stored in a file and is not inherent that such data would be stored in a file. (Appeal Brief, page 7, lines 12 – 14)

Examiner's Response to Argument 2

Gaither teaches "updating a caching profile **in response to arrival of a file**." As discussed above, Gaither's invention collects or updates run-time behavioral information of programs in 'caching profiles'. A program that can be compiled (see figure 1) is stored in one or more files. The run-time caching behavior information of a program (i.e. file) cannot be collected unless the program is running in the computer. The program (a file) must arrive at a computer memory and more specifically at a cache in order for the run-time software to be able to observe and collect information (i.e. updating a caching profile) about its run-time caching behavior. Therefore, "updating a caching profile" is necessarily "**in response to arrival of a file**."

Summary of Appellant's Argument 3

> Gaither does not teach "**responsive to** the act of updating, analyzing information stored in the caching profile".

Examiner's Response to Argument 3

Gaither teaches "**responsive to** the act of updating, analyzing information stored in the caching profile." As discussed above, the run-time behavior information that is gathered is used to determine class attributes. This determination of attributes requires analyses as argued previously. It has also been argued above that the 'cache profile' is updated as the run-time

behavior information is gathered by the software. The analysis of information cannot be performed in the absence of information. Therefore, analyzing is necessarily **responsive to** the act of updating. Also note that "dynamic analysis" implies recursive or reiterative analysis. This suggests that the analysis is responsive to prior updating of the caching profile (i.e. gathering of new information).

Claim 4

Summary of Appellant's Argument

> Since Claim 4 depends from claim 3, which Appellant has argued *supra* to be patentable and in condition for allowance under 35 U.S.C §102(e), Appellant maintains that claim 4 is likewise patentable and in condition for allowance.
>
> Gaither does not teach "wherein the act of analyzing is performed by a predictive modeling engine" as argued *supra* in conjunction with claim 1.

Examiner's Response

> Arguments regarding claim 3 have been address above.
>
> The claimed limitation "wherein the act of analyzing is performed by a predictive modeling engine" has been addressed above in reference to claim 1.

Claim 5

Appellant presents two separate arguments, each of which are summarized and responded to individually.

Summary of Appellant's Argument 1

Gaither does not teach "**responsive to arrival of a file at a cache**, analyzing information stored in a caching profile by computing plurality of metrics".

Gaither does not teach "responsive to arrival of a file at a cache".

Examiner's Response to Argument 1

The first part of the claimed limitation "responsive to arrival of a file at a cache" has been addressed above in reference to claim 3, argument 2.

Summary of Appellant's Argument 2

Claim 5 requires at teaching of at least two metrics.

Claim 5 requires "**computing** a plurality of metrics".

Examiner's Response to Argument 2

Gaither discloses at least three measures, or metrics, that are computed: frequent observation, infrequent observation and residency time (col. 8, lines 42 – 45), that are the bases for determination of likelihood of future use. Each one of these metrics requires computation. Frequency observations, either frequent or infrequent, require running counts, which involve at least an addition. Residency time observations require computations of the time differences between arrivals and departures, which involve subtractions.

Claims 6 and 7

Summary of Appellant's Argument

Gaither does not teach "wherein the plurality of metrics includes clustering metrics".

claim 6 requires that a comparison of metrics is a cause of the selected caching algorithm, whereas the non-uniform distribution in Gaither (which the Examiner alleges to be a clustering metric of claim 6) is not a cause of the selected caching algorithm, but is rather a consequence of the selected caching algorithm.

Gaither does not teach "wherein the plurality of metrics includes scattering metrics".

claim 7 requires that a comparison of metrics is a cause of the selected caching algorithm, whereas the uniform distribution in Gaither (which the Examiner alleges to be a scattering metric of claim 7) is not a cause of the selected caching algorithm, but is rather a consequence of the selected caching algorithm.

Examiner's Response

The claimed limitation is met by Gaither's disclosure in two ways. Firstly, Appellant's specification discloses that metrics for most used and least used caching algorithms are examples of measures of clustering and scattering, respectively (see page 8, lines 4 – 5, and page 10, lines 8 – 11). These are measures of relative frequencies. Gaither discloses these measures in col. 8, line 41 – 44, which in part states "... the cache may be periodically observed to detect entries that are **frequent** (i.e. most used, clustering metric) or **infrequent** (i.e. least used, scattering metric)". These metrics are used to assign attributes that result in selections of algorithms that exhibit non-uniform and other distributions taught in Gaither.

Secondly, the Examiner agrees with Applicant that non-uniform distribution cited in the rejection is a consequence of a caching algorithm selection. However, the Examiner does not

agree with Applicant's allegation that "the Examiner's failure to consider the cause and effect relationships discussed *supra* has resulted in the Examiner incorrectly concluding that Gaither's non-uniform distribution is a metric recited in claim 6." Appellant makes a similar allegation with respect to claim 7. First of all, the Examiner has not characterized the non-uniform and distributions as metrics cited in claims 6 and 7, respectively. Secondly, it is precisely because the Examiner has considered the cause and effect relationship that the Examiner believes that the cited non-uniform and uniform distribution properly reads on the claimed limitations. According to the Applicant's specification, page 7, lines 12 – 14, "The metric computed for each caching algorithm is a figure of merit that suggests how effective the caching algorithm would be if it were applied to present caching situation." In other words, metrics are associated with selected algorithms and are measures of effectiveness of caching behaviors of associated algorithms. Appellant does not offer any other definitions of what these metrics are. The Examiner believes that metrics that are associated with algorithms that exhibit clustering behavior (i.e. non-uniform distribution or uneven distribution, thus clustering) and scattering behavior (i.e. uniform distribution or even scattering) can be properly labeled "clustering metrics" and "scattering metric", respectively.

Claim 9

Summary of Appellant's Argument 1

> Since claim 9 depends from claim 5, which Appellant has argued *supra* to be patentable and in condition for allowance under 35 U.S.C §102(e), Appellant maintains that claim 9 is likewise patentable and in condition for allowance.

Examiner's Response to Argument 1

The Examiner believes that claim 5 is anticipated by Gaither as discussed above.

Summary of Appellant's Argument 2

Gaither does not teach "wherein the act of analyzing is performed by a predictive modeling engine" as explained *supra* in conjunction with claim 1.

Examiner's Response to Argument 2

The Examiner's position on this argument has been presented above in conjunction with claim 1.

Issue 2 – Claim 8 is unpatentable under 35 U.S.C. § 103(a) over Arlitt et al. (US Patent No. 6,272,598) in view of admitted prior art.

Summary of Appellant's Argument 1

Since claim 8 depends from claim 5, which Appellant has argued *supra* to be patentable under 35 U.S.C. § 102(e), Appellant maintains that claim is not unpatentable under 35 U.S.C. § 103(a).

Examiner's Response to Argument 1

This is a spurious argument, since the contested rejection of claim 5 is based on Gaither, not Arlitt.

Summary of Appellant's Argument 2

Arlitt in view of the admitted prior art does not teach or suggest the feature: "responsive to arrival of a file at a cache, analyzing information stored in a caching profile by computing a plurality of metrics".

Arlitt is totally silent as to the "responsive to arrival of a file at a cache" aspect of the preceding feature of claim 8.

Arlitt is totally silent as to "information stored in a caching profile".

Examiner's Response to Argument 2

Appellant quotes the sentence cited in the rejection, "The hit rate and byte hit rate are two main (or most common) performance metrics used to measure performance of proxy cache such as the cache 72" and concludes that Arlitt does not teach the claimed limitation. Contrary to Appellant's erroneous conclusion, Arlitt teaches every limitation that is being contested. First of all, "arrival of a file at a cache" is taught since this is inherent in Arlitt's invention. The subject matter of Arlitt's invention is Web caching (see title). Arlitt's Web cache caches Web documents (see col. 1, lines 15 – 29) that are composed of various files. Each of the files is referred to as an "object" in Arlitt's disclosure (col. 1, lines 23 – 24). The Web cache caches these objects. Web page files must arrive at and reside in the cache in order for Arlitt's invention to be functional. Secondly, "responsive to arrival ..." is taught as well. In the art of caching, a cache hit occurs when a requested object is present in the cache. A cache miss is declared when the requested object is not is in the cache (see col. 1, lines 53 – 54). Hit rates cited above cannot be determined in the absence of an object or file in the cache. Determination of hit rates can only be conducted in response to arrival of a file, since a reference to the file prior to its arrival is classified as a "miss".

Appellant further argues that Arlitt does not disclose "information stored in a caching profile." Again, this limitation is taught by Arlitt as well. Measurements of hit rate and byte hit rate require keeping track of at least the number of hits and byte hit counts, which in turn require storage of at least the running counts of hits and bytes information. Measurement of a "rate" requires computation.

Summary of Appellant's Argument 3

Arlitt does not teach "responsive to a comparison of the metrics one with another".

Examiner's Response to Argument 3

The passage in Arlitt reference cited in the rejection, col. 5, lines 35 – 47, discloses that the cache manager uses multiple replacement strategies (caching algorithms). Arlitt specifically states that "Each **replacement policy** is optimized for one **performance metric**." Arlitt makes it clear that the use of a particular replacement policy (caching algorithm) is based on the performance metric. How would a cache manager select a replacement strategy from a number of replacement strategies, each of which are keyed to one performance metric, without examining and comparing the associated metrics when they are the bases for the selection of the replacement strategy to be used? The metrics must be compared in order for Arlitt's invention to function as described.

Summary of Appellant's Argument 4

> Appellant acknowledges that the specification of the present patent application states that a most-recently used caching algorithm and a most-used algorithm are known in the art. However, the admitted prior art doe disclose selecting (in comparison of the metrics one with the another) a preferred caching algorithm from a plurality of caching algorithms that include a most-recently-used caching algorithm and a most-used algorithm, as required by claim 8. Therefore, the issue of whether the admitted prior can be combined with Arlitt is not relevant, since the admitted prior art does not teach or suggest the aforementioned feature of claim 8.

Examiner's Response to Argument 4

Appellant's argument is confusing and inconsistent. On the one hand, Appellant states that admitted prior art does disclose a certain feature of the claim (Brief, page 17, 16 – 19). In the next sentence, Appellant states that admitted prior art does not teach or suggest aforementioned feature of the claim.

Not withstanding the inconsistency in Appellant's argument, the argument is not persuasive since it is solely based on one of the prior art references not teaching (or teaching) a specific claimed limitation. The test for obviousness is not that the claimed invention must be expressly suggested in any one or all of the references. Rather, the test is what the combined teachings of the references would have suggested to those of ordinary skill in the art. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981).

As discussed in the rejection of the claim, selecting a preferred caching algorithm from a plurality of caching algorithms is taught by Arlitt. Prior art teaches additional algorithms that Arlitt does not specifically disclose. It is the teaching of the additional algorithms that is being combined with Arlitt's invention. Motivation to combine the teachings is clearly stated in the

rejection. One would be motivated to incorporate additional cache replacement algorithms to increase the versatility and enhance the overall performance of the cache since the additional algorithms would allow the cache to adapt to caching patterns that may not have been handled in an optimal way because fewer caching options were available prior to the combination. Also note that Arlitt specifically discloses that other known cache replacement algorithms may also be used (col. 7, lines 28 – 29).

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Woo H. Choi
Assistant Patent Examiner
February 20, 2004

Conferees

David Robertson
Primary Patent Examiner
February 20, 2004

Mathew Kim
Supervisory Patent Examiner
February 20, 2004

David R. Irvin
IBM Corporation T81/503
PO Box 12195
Research Triangle Park, NC 27709